REMARKS

The present Amendment is responsive to the Office Action mailed June 30, 2005.

In the Office Action, claims 1-25, 71-90 and 94-96 were rejected as being anticipated by Kido et al. Reconsideration and withdrawal of these rejections are respectfully requested.

As the Examiner will note, the claims have been amended to more precisely define the claimed embodiments and to take care of a few formal housekeeping matters. No new matter has been added.

Each of the pending independent claims is discussed hereunder in turn.

Independent Claim 1

Independent claim 1, as amended, recites:

> **1. (Currently Amended) A PKI certificate architecture for a network connected gaming system, wherein each software component within the gaming system subject to receive certification is uniquely associated with a unique identifier and is signed with a ~~distinctive~~ separate and unique PKI certificate, the separate and unique PKI certificate being uniquely identified ~~by at least one field~~ by the unique identifier.**

Therefore, independent claim 1 requires that each software component to receive certification is uniquely associated with a unique identifier and that each software component is signed with a separate and unique certificate that is, in turn, uniquely identified by the unique identifier. According to claim 1, a separate and unique certificate is created for each software component subject to receive certification. This separate and unique certificate, in turn, is identified at least by the unique identifier (its part number, for example) that is associated with (or part of, for example) its software component.

In contrast, Kido et al. teach that each execution object 121 has a certificate 210 and an electronic signature 221 **of its provider**. (See Col. 11, lines 55-65 identified by the Examiner). Therefore, Kido et al. teach that each execution object has a certificate and an electronic signature

REMARKS

of its provider (i.e., the entity that provided the execution object - in Kido et al.'s case, the assignee IBM, for example).

In Kido et al., the certificates are evaluated to determine whether they originate from the "proper provider." See, e.g., the claims of Kido et al., Col. 12, line 47 to Col. 13, line 10 ("...an electronic signature 271 of the provider, which represents data encrypted using the secret key of the provider, enables to confirm that the data is not corrupted since it has been created by the provider." In Kido et al., therefore, the certificates and the code signing operations rely upon the secret key of the provider, and are not specific to any software component. Kido et al. are concerned about the proper provenance of the data or execution objects (i.e., where did the data or execution objects originate from), whereas the claim 1 calls for each software component to receive regulatory certification have a "separate and unique certificate" that is "uniquely identified at least by a unique identifier that is uniquely associated with the software component (i.e., there is a one-to-one correspondence between the unique identifier and the software component). According to claim 1, therefore, each software component receives a separate and unique certificate that is uniquely identified by a unique identifier that is uniquely associated with the software component. Kido et al. do not teach this. Indeed, there is no teaching in Kido et al. of each software component having its own separate and unique certificate, and much less a PKI certificate architecture in which each software component has its own separate and unique certificate that is uniquely identified (at least) by a unique identifier that is uniquely associated with the software component. Each certificate, according to the claimed inventions, therefore, is separate and unique, with the exception of the embodiment that includes companion files (discussed hereunder relative to independent claim 94). In Kido et al., the certificates appear to be unique only to the provider of

REMARKS

the execution objects, and not to the executable objects themselves, as required by claim 1 and its dependent claims.

In Kido et al., there is no link, correspondence or unique association between a unique identifier of their execution objects and the certificates. In Kido et al., the certificates are created with the secret key of the provider of the execution objects, whereas, the claimed embodiments call for a separate and unique certificate to be created for each software component, and this separate and unique certificate is uniquely identified by a unique identifier (the part number thereof, for example) that is uniquely associated with software component. Claim 1, as amended, makes this distinction clear and unambiguous. Unique identifiers, according to further embodiments of the present invention, may include any one or a combination of a software component part number, major version number; a software component minor version number; a software component build number; a software component revision number; a software component project name; a software component type of software component; a software component language variant; a software component game regulation variant; a software component friendly name; an identification of the certification laboratory, and/or an identification of the client, as called for by claim 7.

For the Examiner's convenience, Applicants' representative has reproduced the entire text of Kido et al. in the APPENDIX attached hereto, highlighting all instances of the term "provider" to emphasize the point that all certificates in Kido et al. are encrypted with the secret key of the provider, and are not uniquely identified by a unique identifier that is uniquely associated with the software component, as claimed and required by claim 1.

Independent Claim 17

Independent claim 17, as amended, recites:

REMARKS

> **17. (Currently Amended) A method for a network connected gaming system to prevent unauthorized software components of constituent computers of the gaming system from executing, comprising the steps of:**
>
> **producing a separate and unique PKI certificate for each software component subject to receiving certification, each software component subject to receiving certification including a unique identifier;**
>
> **code signing each software component subject to receiving certification with its respective separate and unique PKI certificate, each respective PKI certificate being uniquely identified by a unique identifier that is uniquely associated with the software component, and**
>
> **configuring ~~Software Restriction Policy~~ software restriction policy certificate rules to allow execution of ~~a selected set of each software component subject to receiving certification~~ only those software components whose code signed PKI certificate is determined to be authorized.**

As the Examiner will note, independent claim 17 calls for producing a separate and unique PKI certificate for each software component (Kido et al. do not do this – their certificate attests to the "proper provider" and is not unique to any one software component). Moreover, each certificate, as claimed, is uniquely identified by a unique identifier that is uniquely associated with the software component. Again, the certificates of Kido et al. even if separate for each execution object, is not unique to each execution object – instead, it refers to the common provider thereof, which is hardly unique.

The discussion above relative to claim 1 is equally applicable to independent claim 17 and is incorporated herein by reference, as if repeated here in full.

Independent claims 20 and 22

Independent claim 20, as amended, recites:

> **20. (Currently Amended) A method for a network connected gaming system to enable only authorized software components of constituent computers of the gaming system to execute, comprising the steps of:**
>
> **configuring a separate ~~Software Restriction Policy~~ software restriction policy for each authorized software component, and**
>
> **enforcing the ~~Software Restriction Policy~~ software restriction policy for each authorized software component.**

REMARKS

It is respectfully submitted that the passages identified by the Examiner as teaching the claimed steps (Col. 11, line 55 to Col. 12, line 12 and Col. 12, line 46 to Col 13, line 10) do not teach any software restriction policy. Moreover, these passages do not teach that a separate software restriction policy is configured and enforced for each authorized software component, as claimed herein. Again, Kido et al. in Col 12, lines 51-59 state that the execution objects are provided with the electronic signature of the provider, to insure that the file has not been corrupted since its creation by the provider. Independent claim 22 also includes a similar limitation drawn to a separate and unique software restriction policy for each authorized software component, which finds no counterpart in the Kido et al. reference.

Independent Claim 24

As amended, claim 24 recites:

> **24. (Currently Amended) A method for a network connected gaming system to enable only authorized software components of constituent computers of the gaming system to execute, comprising the steps of:**
>
> **producing a separate and unique PKI certificate for each software component subject to receive certification, each respective PKI certificate being associated with a unique identifier that is uniquely associated with the software component;**
>
> **signing each software component subject to receive certification with ~~the~~ its respective separate and unique PKI certificate;**
>
> **configuring a certificate ~~Software Restriction Policy~~ software restriction policy for each of the respective separate and unique PKI certificates, and**
>
> **enforcing the certificate ~~Software Restriction Policy~~ software restriction policy for each of the respective separate and unique PKI certificates.**

Independent claim 24, therefore, includes recitations drawn to a separate and unique PKI certificate for each software component subject to regulatory certification, as well as recitations drawn to a separate and unique software restriction policy for each software component. Therefore, claim 24 is allowable for all of the reasons developed above, as Kido et al. do not teach a separate and unique certificate for each software component, nor do Kido et al. teach the creation or enforcement of a separate software restriction policy for each software component.

REMARKS

Independent claim 25

As amended, independent claim 25 recites:

> **25. (Currently Amended) A method for downloading authorized software components and allowing execution of downloaded authorized software components ~~for~~ of constituent computers of a network connected gaming system, comprising the steps of:**
>
> **code signing each authorized software component with a ~~distinctive~~ separate PKI certificate that is unique to the authorized software component;**
>
> **packaging the code signed authorized software components into an installation package;**
>
> **configuring install policies to install each code signed authorized software component contained in the installation package;**
>
> **configuring certificate rule policies to allow execution of the installed code signed authorized software component;**
>
> **configuring enforcement of the policies.**

As developed above, Kido et al. teach to associate a certificate to the execution object that is unique to the provider of the execution object, and not a certificate that is unique to the execution object itself. In contrast, as claimed herein, the authorized software component is code signed with a separate PKI certificate that is unique to the authorized software component, which is clearly not taught by the Kido et al. reference, as examination of the text thereof in the Appendix reveals.

Independent claims 71 and 73

As amended herewith, claim 71 recites:

> **71. (Currently Amended) A method for a network connected gaming system to prevent unauthorized executable files of constituent computers of the gaming system from executing, comprising the steps of:**
>
> **packaging the authorized executable files into a code signed ~~MSI~~ installation package;**
>
> **configuring certificate rule policies to enable execution of the code signed ~~MSI~~ installation package;**
>
> **enforcing the policies, and**
>
> **executing the code signed ~~MSI~~ installation package upon every ~~computer~~ startup of any of the constituent computers of the gaming system or upon a command, wherein execution of any authorized executable file is predicated upon successfully executing the code signed installation package into which the authorized executable file is packaged.**

REMARKS

Claim 71 calls for packaging authorized executable files into a code signed installation package, configuring and enforcing certificate rule policies and executing the code signed installation package upon every startup of any of the gaming machines of the gaming system. At the outset, there is no teaching of any execution of any installation package upon every startup of gaming (or any other) machines. This shortcoming alone warrants withdrawal of the anticipation rejection. Moreover, claim 71 requires that execution of any authorized executable file is predicated upon successfully executing the code signed package into which the authorized executable file is packaged. Independent claim 73 contains similar recitations. None of the passages pointed out by the Office even hint at such steps or functionality, nor does the remainder of this reference. Note that none of Kido et al.'s Figs 6-8 nor the corresponding written description thereof teach such steps or functionality.

Independent claims 75 and 77

As amended, claim 75 recites a method to prevent code of unauthorized **non-executable** files from affecting the game outcome, and includes steps of:

> **75. (Currently Amended) ...**
> **packaging the non-executable files into a code signed ~~MSI~~ installation package;**
> **configuring certificate rule policies to enable execution of the code signed ~~MSI~~ installation package;**
> **configuring enforcement of the policies, and**
> **executing the code signed ~~MSI~~ installation package upon every ~~computer~~ startup of any of the constituent computers of the gaming system or upon a command.**

In this claimed embodiment, the non-executable files are packaged into a code signed installation package, certificate rules are configured and enforced, and the code signed installation package is executed upon every startup of any of the computers (e.g., gaming machines, gaming terminals, gaming servers, payment terminals, etc.) of the gaming system or upon command. Kido

REMARKS

et al. include no teachings of packaging non-executable files into an installation package, nor does Kido et al. teach of executing an installation package upon every startup of any computer or upon command, as claimed. Claim 77 contains similar limitations, in that the installation package is re-installed upon every startup of any computer, a topic about which Kido et al. is also wholly silent.

Independent claim 79:

> **79. (Currently Amended) A method for scheduling at least one authorized executable software component installed in a network connected gaming system, the gaming system including a plurality of gaming machines, the method comprising the steps of:**
>
> **packaging at least one authorized non-executable file that control the scheduling of the at least one authorized executable software component into at least one code signed ~~MSI~~ installation package, each of the at least one code signed installation packages including a predetermined PKI certificate;**
>
> **configuring certificate rule policies to enable execution of the at least one code signed ~~MSI~~ installation package in ~~a selected set of gaming terminals~~ selected ones of the plurality of gaming machines; and**
>
> **configuring enforcement of the certificate rule policies; and**
>
> **downloading the at least one code signed ~~MSI~~ installation package into ~~a selected set of gaming terminals~~ the selected ones of the plurality of gaming machines;**
>
> **executing the at least one code signed ~~MSI~~ installation ~~packages~~ package.**

At the outset, Kido et al. do not teach any method for scheduling anything (the words schedule, scheduling and the like do not even appear in the patent). Neither Fig. 3 of Kido et al. nor the passages of the written portion thereof referred to in the outstanding Office Action teach any method or means for scheduling authorized executable software components or anything else. Moreover, Kido et al. do not teach that each of the installation packages (Kido et al. do not teach any installation packages, or otherwise) includes a predetermined PKI certificate.

Independent claim 82

As amended, claim 82 recites:

REMARKS

82. (Currently Amended) An automated platform to enable ~~the~~ an on-going regulatory certification of a ~~substantial number~~ plurality of authorized software components of a network connected gaming system including a plurality of computers, the method comprising:

a reference platform representative of a target network connected gaming system and comprising a software-building environment located at a manufacturer or subcontractor of the software components ~~the manufacturer's premises or designated subcontractors~~;

a certification platform located at a regulatory certification authority, the certification platform being substantially identical to the reference platform, and

code-signing means for enabling the manufacturer or subcontractor to associate a ~~associating a distinctive~~ separate and unique PKI certificate with each authorized software component subject to regulatory certification.

The passages identified in the Office Action as teaching the subject matter of claim 82 do not, in fact, do so. For example, Col. 11, lines 20-35 is merely a listing of the different operating systems with which the Kido et al. system is compatible. Col. 11, lines 40-45 discusses, among other items, the object server 150. The object server is described in Kido et al. beginning at Col. 12, line 64 as follows:

> **The object server 150, which is managed by the associated provider, is responsive to a request from the stub object 110 for notifying term's validity (guaranteed term of availability) of each version of execution objects. Also, the object server 150 contains the newest execution objects, thereby allowing the stub object 110 to access the same. Referring to FIG. 5, the object server 150 has an execution object management table 300, which contains execution objects' identifiers 301, term's validity information 303 of respective versions of the execution objects and pointers 305 to the execution objects.**

The object server, therefore, acts as a repository for execution objects and is further tasked with verifying the validity of execution objects. Also, the Examiner's kind attention is drawn to Col. 13, beginning at line 22 under the heading "C. Checking/Referencing of Execution Object." Neither the object server nor the remote systems 141, 145 (see Fig. 3 of Kido et al.) meet the claim recitations of claim 82 relative to the claimed reference and certification platforms. Moreover, Kido et al.'s execution object server and remote systems do not teach any reference platform having code-signing means for enabling the manufacturer or subcontractor to associate a

REMARKS

associating a distinctive separate and unique PKI certificate with each authorized software component subject to regulatory certification, as also claimed in independent claim 82, for the reasons outlined above relative to other independent claims. Kindly note that the Certification Authority (CA) 170 in Kido et al. is only the authority that issues the certificates and is not described by Kido et al. as having any of the functionality of the claimed reference or certification platforms.

Independent claim 94

Lastly, independent claim 94, as amended, recites:

> **94. (Currently Amended) A method for a gaming ~~terminal~~ machine in a network connected gaming system to generate a ~~list~~ menu of authorized games available to players, the method comprising the steps of:**
>
> **generating a separate and unique code signed PKI certificate for a predetermined software module of each authorized game;**
>
> **generating an executable companion file for each authorized game, wherein the executable companion file is configured to execute faster than the authorized game ~~substantially quicker to execute than starting execution of the game and, wherein the code signed PKI certificate associated to the companion file is identical to the code signed PKI certificate associated to the game main module~~;**
>
> **code signing both the predetermined software module and its executable companion file with the generated PKI certificate;**
>
> **enforcing ~~Software Restriction Policy~~ software restriction policy rules for preventing non-authorized software components from executing;**
>
> **enforcing ~~Software Restriction Policy~~ software restriction policy rules for enabling execution of ~~a selected set~~ selected ones of the authorized games;**
>
> **attempting to execute each executable companion file, and**
>
> **adding only those games to ~~a~~ the menu ~~list~~ of authorized games whose executable companion file has not been denied execution by the software restriction policy rules.**

None of the passages referenced by the Examiner teach any menus or methods for generating menus, as required by the claim. In fact, there are no menus disclosed in Kido et al., and there are no methods of generating menus of authorized games available to players in this reference. Moreover, none of the passages referenced by the Examiner or the remainder of Kido et al. even remotely teach the use of a "companion file" that executes faster than its associated game

REMARKS

to determine whether a game should be added to a menu of available games. The concept and implementation of companion files is only described in the present application. Indeed, the embodiments of the present inventions that use a companion file are disclosed in the specification beginning at paragraph [122]. Kido et al. simply teach nothing of the sort.

In view of the present Amendment and the foregoing remarks it is believed that the rejections of the claims under 35 U.S.C. §102(e) should be reconsidered and withdrawn. The same is, therefore, respectfully requested.

Applicants' attorney believes that the present application is now in condition for an early allowance and passage to issue. If any unresolved issues remain, the Examiner is respectfully invited to contact the undersigned attorney of record at the telephone number indicated below, and whatever is required will be done at once.

Respectfully submitted,

Date: Dec 22, 2005

By: _____
Alan W. Young
Attorney for Applicants
Registration No. 37,970

YOUNG LAW FIRM, P.C.
4370 Alpine Rd., Ste. 106
Portola Valley, CA 94028
Tel.: (650) 851-7210
Fax: (650) 851-7232

\\Ylfserver\ylf\CLIENTS\IMG\CYBS\5858 (Trusted Game Download)\5858 AMEND.1.doc